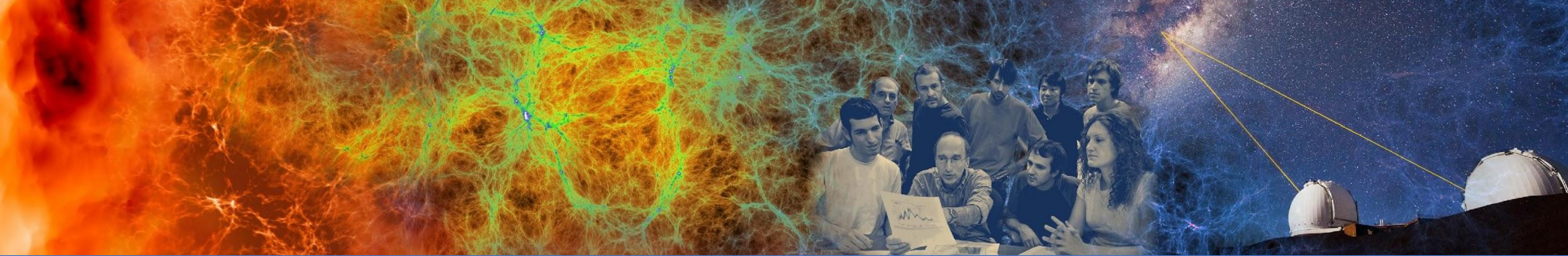
Perlmutter system overview

ATPESC 2021
Track 1 - Hardware Architectures
2021 August 2

Brian Friesen
National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

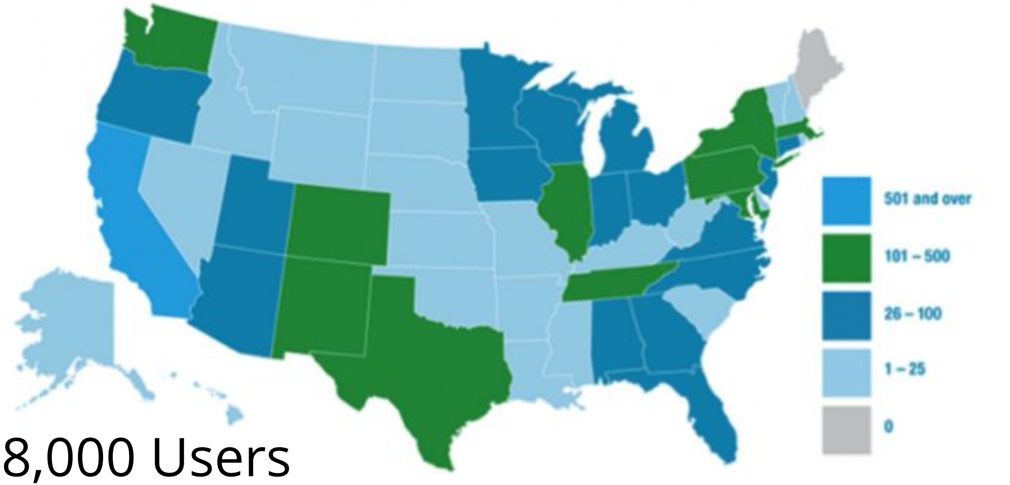
Summary

- Perlmutter is a heterogeneous CPU+GPU system designed to accelerate the diverse data-centric and computational workflows for thousands of NERSC users
- First phase of Perlmutter with all ~6000 NVIDIA A100 GPUs has been delivered in NERSC's data center and is undergoing integration and testing
- The system will support a wide range of programming languages and models, ensuring that its broad workload will be able to use its GPUs effectively

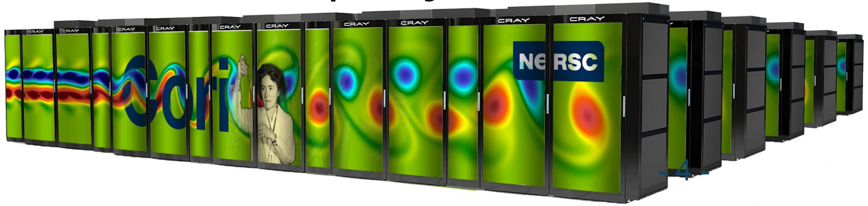


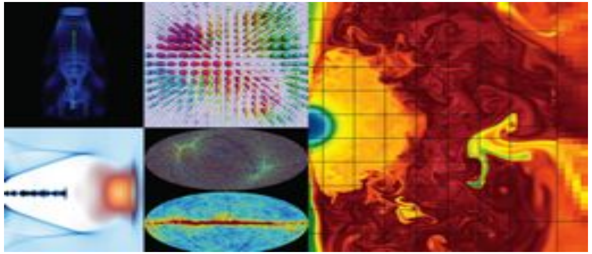
NERSC mission

NERSC is the mission computing facility for the DOE Office of Science

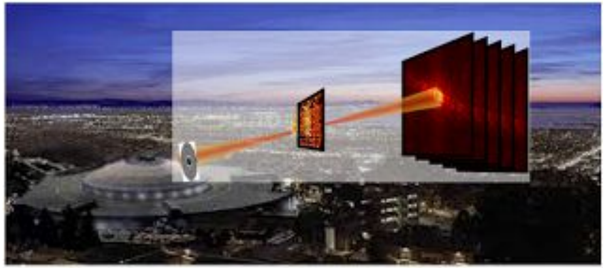


8,000 Users
800 Projects
700 Codes
2000 NERSC citations per year



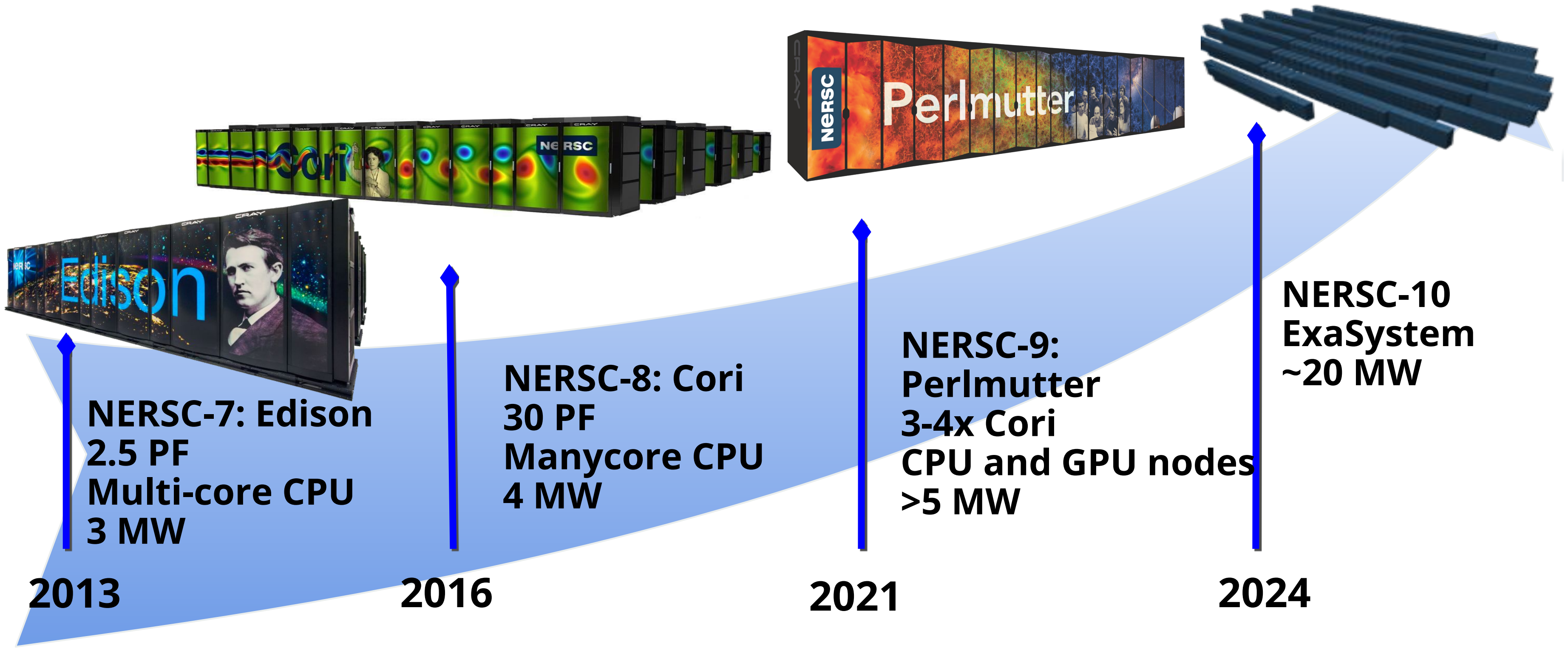


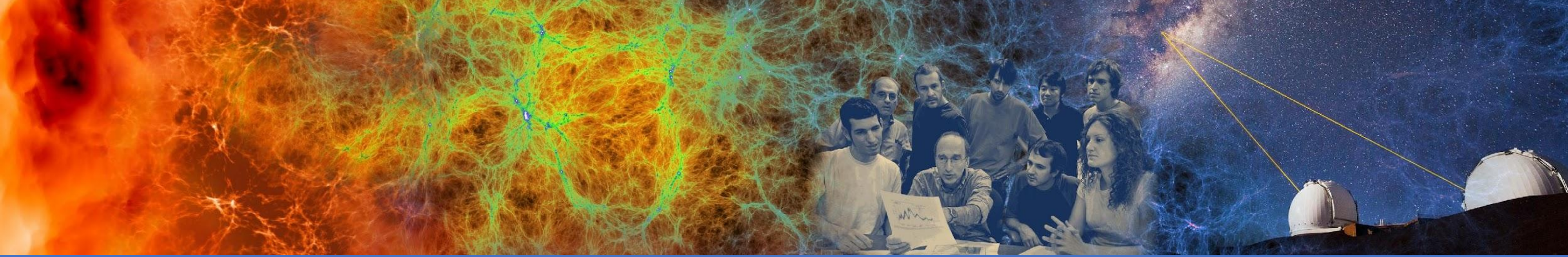
Simulations at scale



Data analysis support for DOE's experimental and observational facilities
Photo Credit: CAMERA

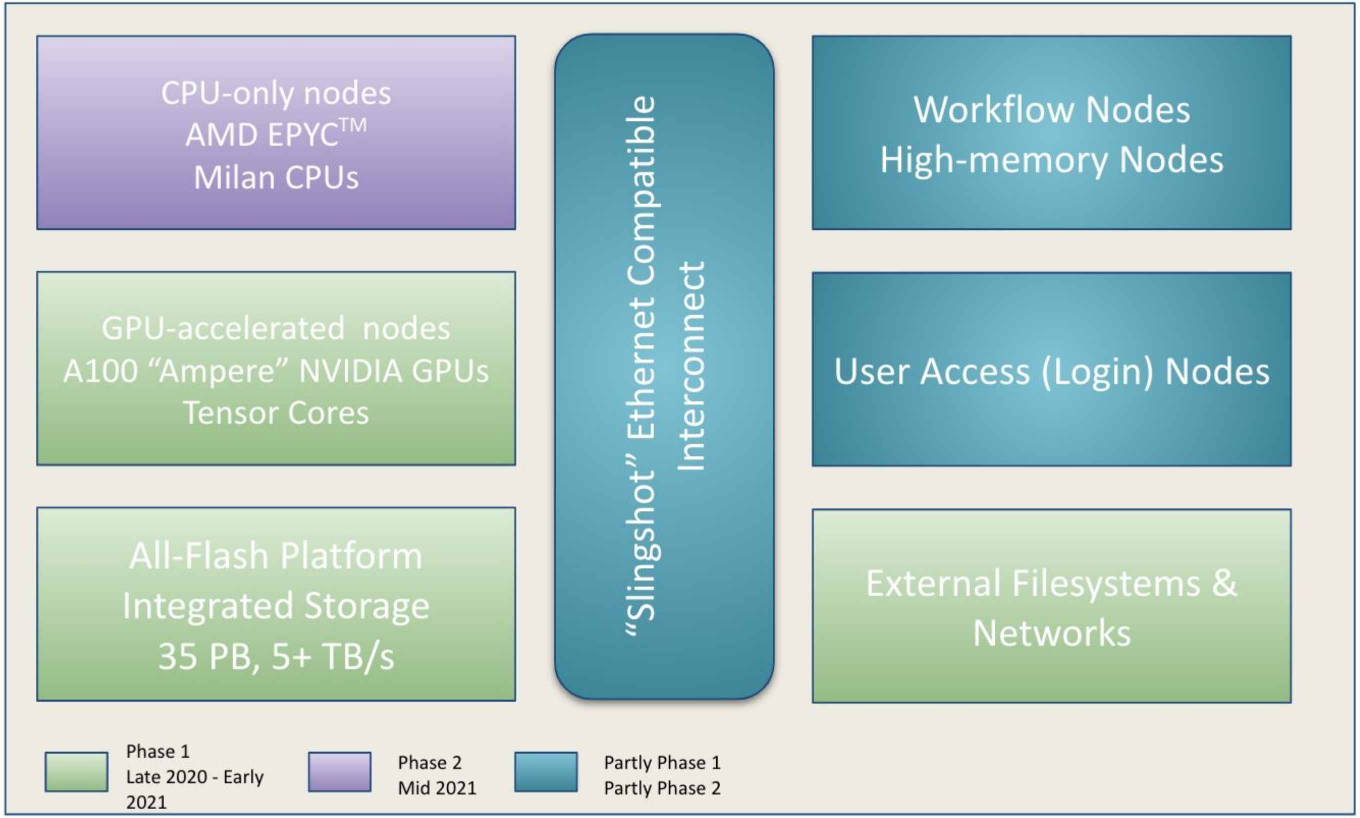
NERSC systems roadmap

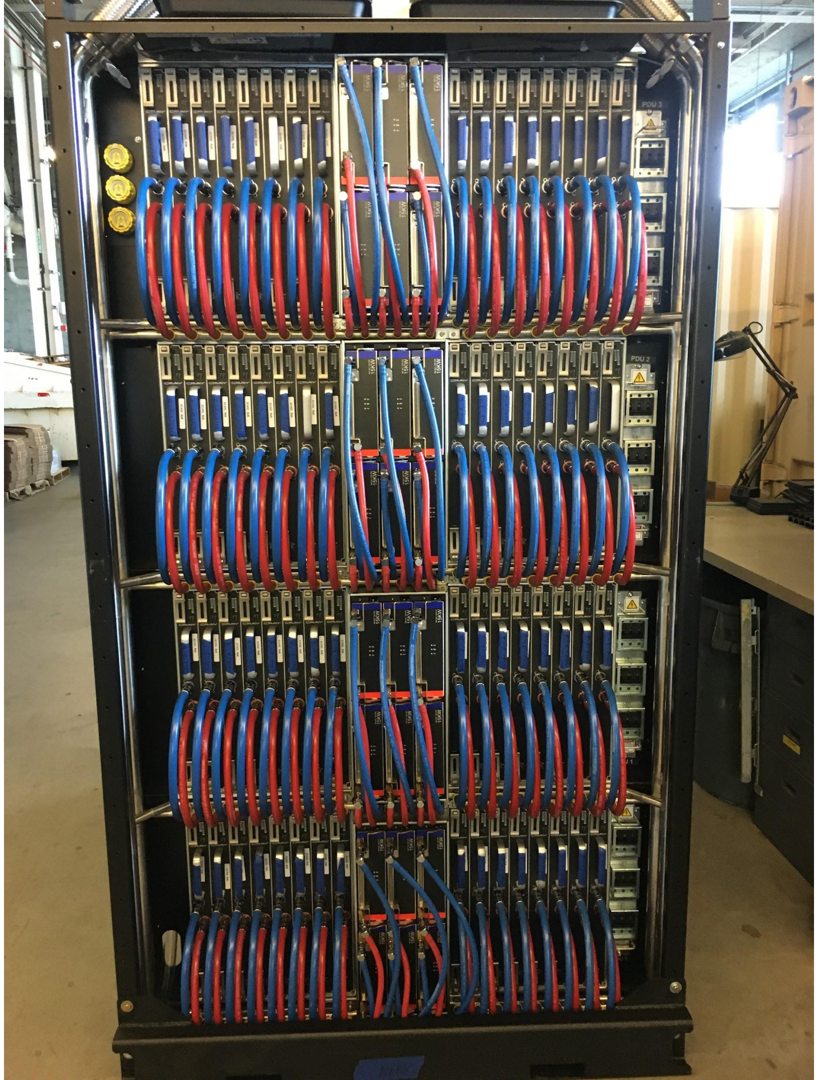


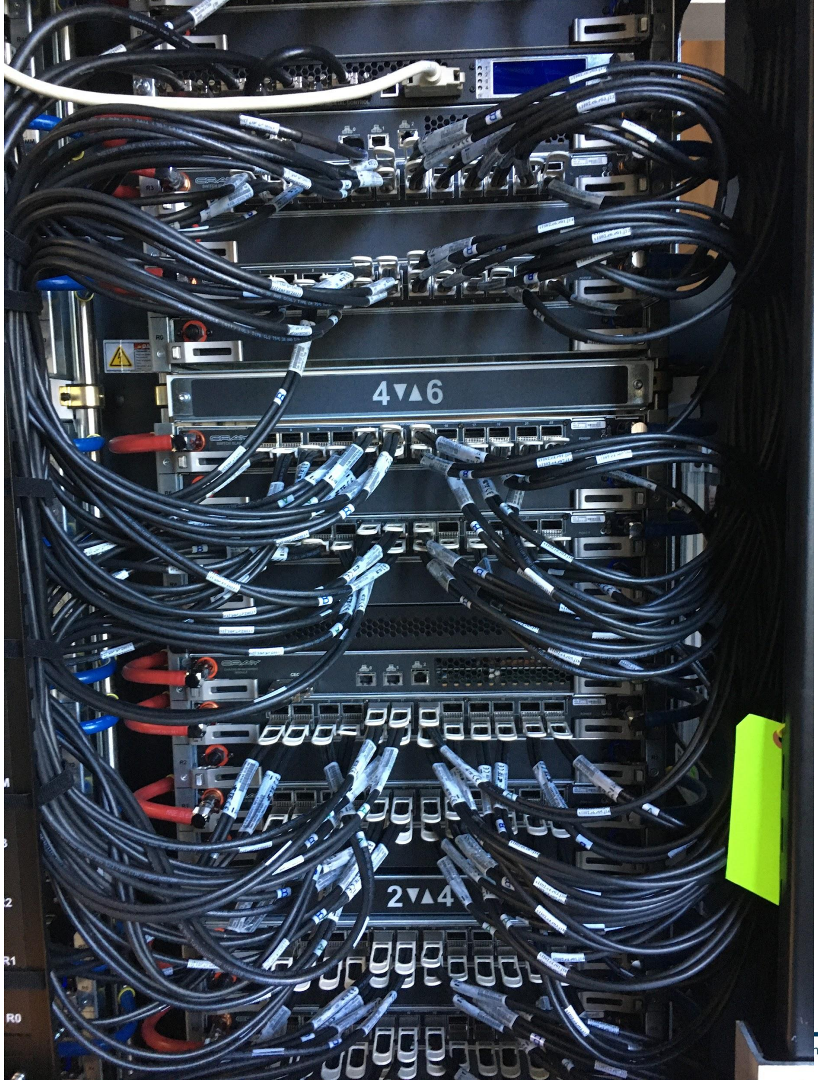


Perlmutter hardware

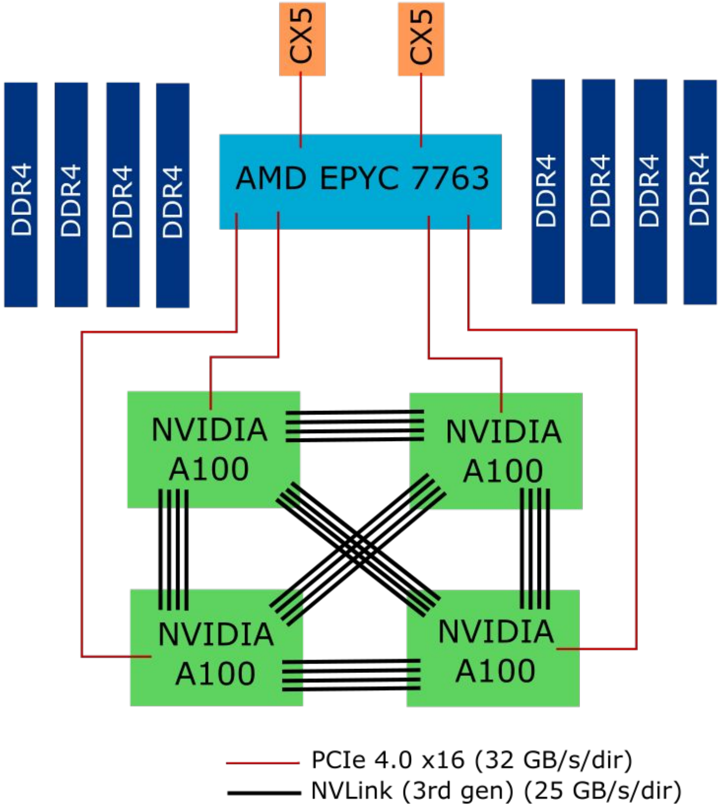
Hardware overview



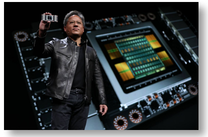



4▼▲6
2▼▲4

GPU nodes in Perlmutter Phase 1



GPU nodes in depth



- 4x NVIDIA A100 GPUs
- 1 AMD EPYC 7763 CPU
- CPU connected to GPUs via PCIe 4.0
- NVLink connected A2A across GPUs, 4x bonded
- FP16, TF32, FP64 tensor cores on GPUs
- Multi-Instance GPU

	V100	A100
FP64 Peak	7.5 TF FMA	19.5 TF TC (9.7 TF FMA)
FP16 Peak	125 TF TC	312 TF TC
SMs	80	108
Memory BW	900 GB/s	1555 GB/s
Memory Size	16 GB (HBM2)	40 GB (HBM2)
L2 Cache	6 MB	40 MB
Shared Mem. / SM	96 KB	164 KB

High-speed network

- Perlmutter's high-speed network ("Slingshot") connects compute nodes, Lustre storage nodes, login nodes, workflow nodes, and other service nodes into a single network
- Uses a "dragonfly" topology, and is an evolution of the dragonfly topology used in Cray's XC product
- Dragonfly is a hierarchy of A2A connections, with each group acting as a as a "virtual" high-radix router, to minimize the diameter of the network while also minimizing the use of active optical cables (which are very expensive)

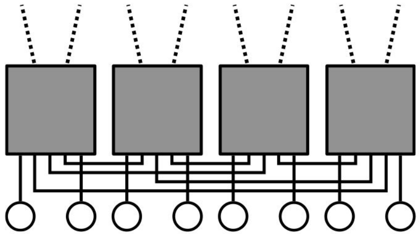
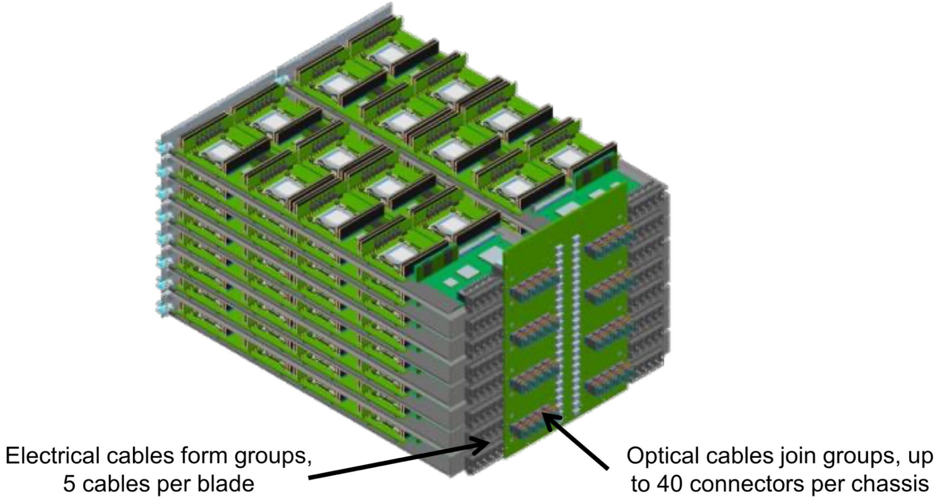


Fig. 1: A dragonfly group with all-to-all local connections. Boxes are routers, circles are nodes, solid lines are electrical local links, and dashed lines are optical global links.

Kaplan, et al. 2017. "Unveiling the Interplay Between Global Link Arrangements and Network Management Algorithms on Dragonfly Networks." In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (CCGrid '17). IEEE Press, 325–334.

Aries HSN on Cray XC systems

- Each compute blade has 4 nodes and 1 Aries ASIC; 16 blades form a chassis
- Each of the 16 Aries ASICs in a chassis are connected A2A with a backplane
- Each chassis (3 per cabinet) is connected A2A to the other 5 chassis in a 2-cabinet group with electrical cables (short, fast, cheap)
- Each group connected A2A to every other group with active optical cables (long, fast, expensive)
- Aries HSN protocol is specialized and proprietary - only special types of nodes can use it



Faanes et al., 2012. Cray Cascade: A scalable HPC system based on a Dragonfly network. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC '12).* IEEE Computer Society Press, Washington, DC, USA, Article 103, 1–9.

Slingshot HSN on Shasta systems

- Compute blades and Slingshot switch blades are oriented at 90 deg
  - Compute blades are vertical, switch blades horizontal
  - Enables A2A connectivity to compute nodes in a group without a backplane
- Slingshot is Ethernet-compatible - enables high-bandwidth connectivity to many types of network endpoints, not just compute nodes and specialized service nodes
- Lustre storage nodes connected directly to HSN - no more LNet routers

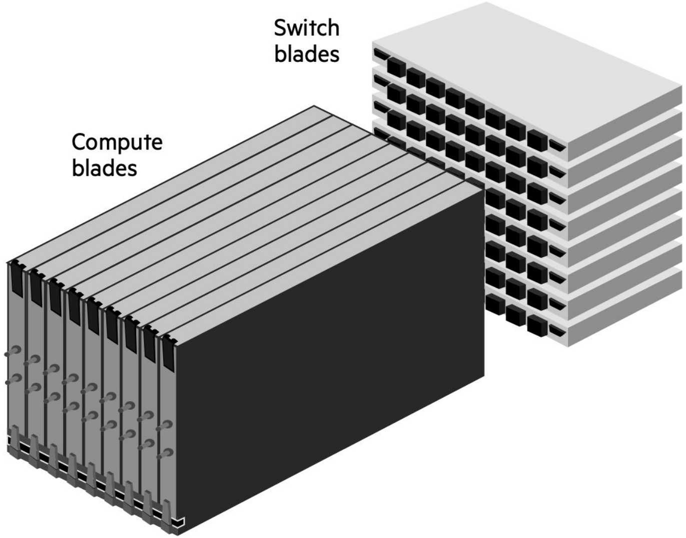
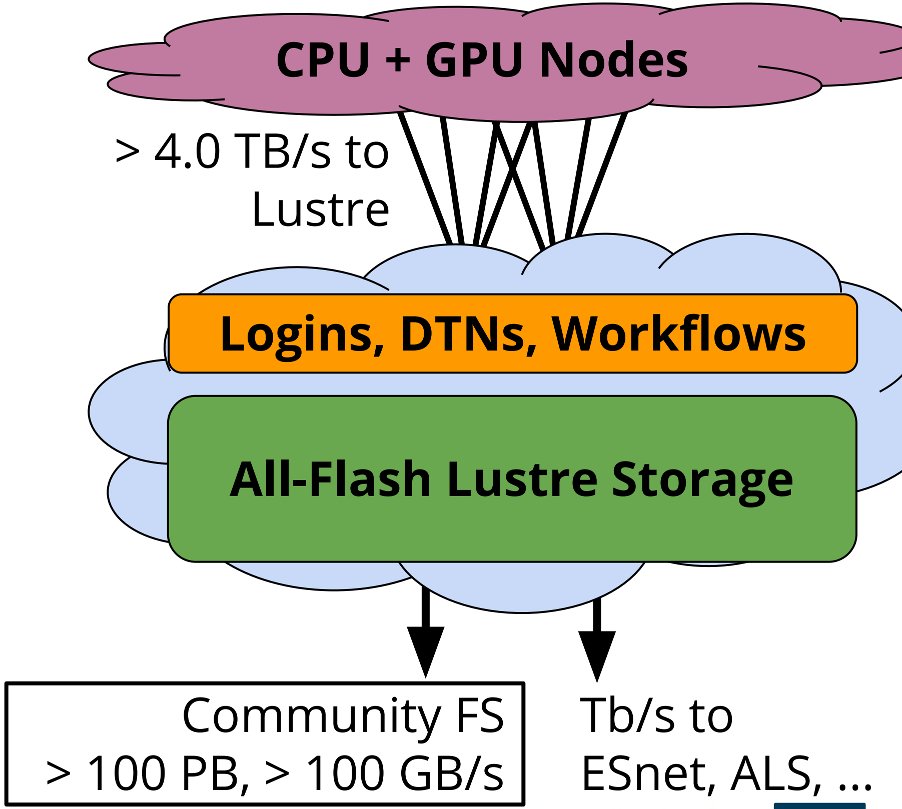


FIGURE 2. Compute blade and switch blade interface

"HPE Cray EX Liquid-Cooled Cabinet for Large-Scale Systems brochure" (hpe.com)

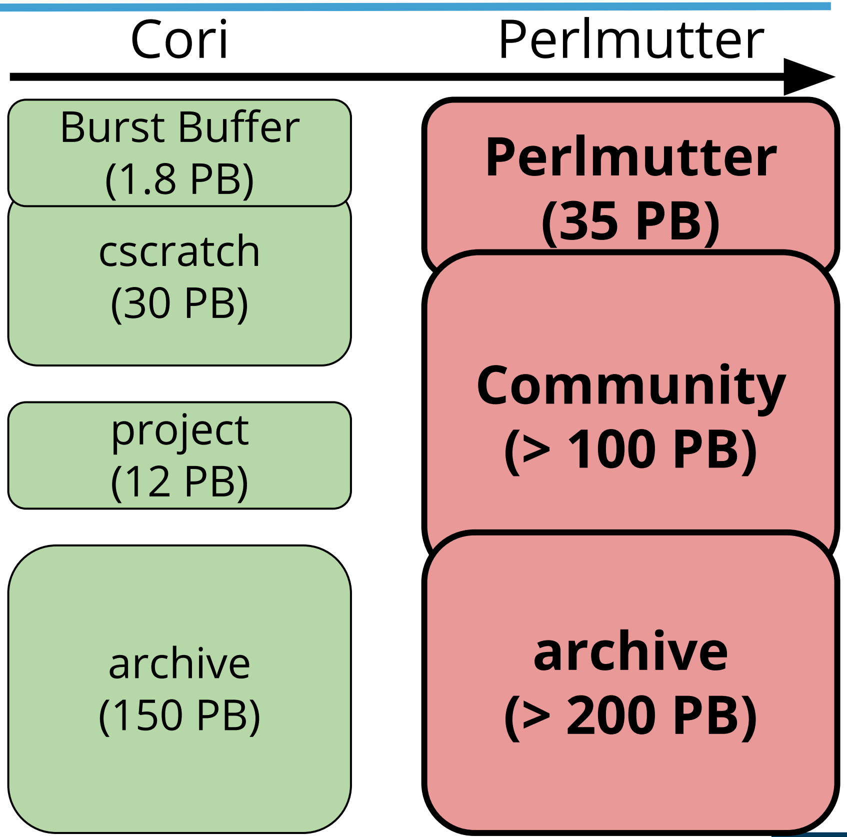
All-flash file system

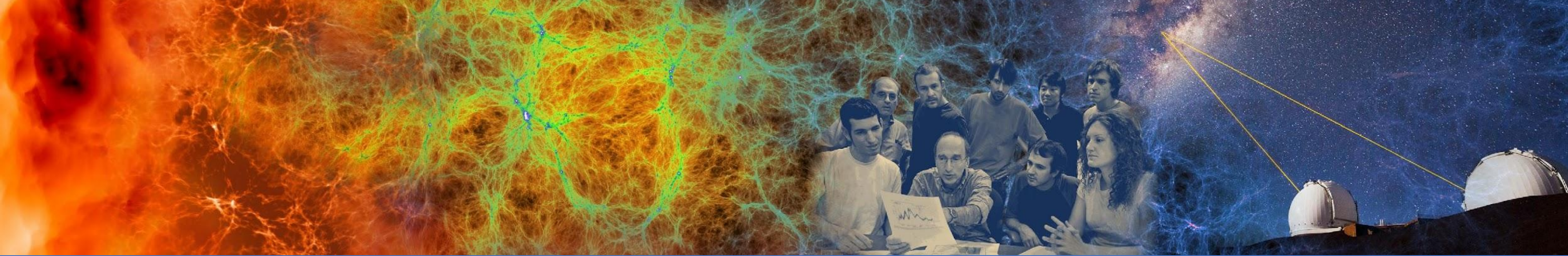
- **<u>Fast</u> across many dimensions**
  - > 4 TB/s sustained bandwidth
  - > 7,000,000 IOPS
  - > 3,200,000 file creates/sec
- **<u>Usable</u> for NERSC users**
  - > 30 PB usable capacity
  - Familiar Lustre interfaces
  - New data movement capabilities
- **<u>Optimized</u> for data workloads**
  - NEW small-file I/O improvements
  - NEW features for high IOPS, non-sequential I/O



Data Movement

- Project file system replaced with Community File System
- NERSC-HPE collaboration will simplify data motion between Perlmutter & CFS
- Bandwidth and capacity are competing resources - tiered storage enables NERSC to spend $ on each where it is most critical





Software configuration and user environment

Perlmutter Programming Environments

	GPU Support	Fortran/ C/C++	OpenACC 2.x	OpenMP 5.x	CUDA	Kokkos, RAJA	Cray MPI	HIP	DPC++ / SYCL
NVIDIA									
CCE									
GNU				(Community Effort)					
LLVM				(Community Effort)					

Vendor Supported

NERSC Supported

GPUDirect RDMA

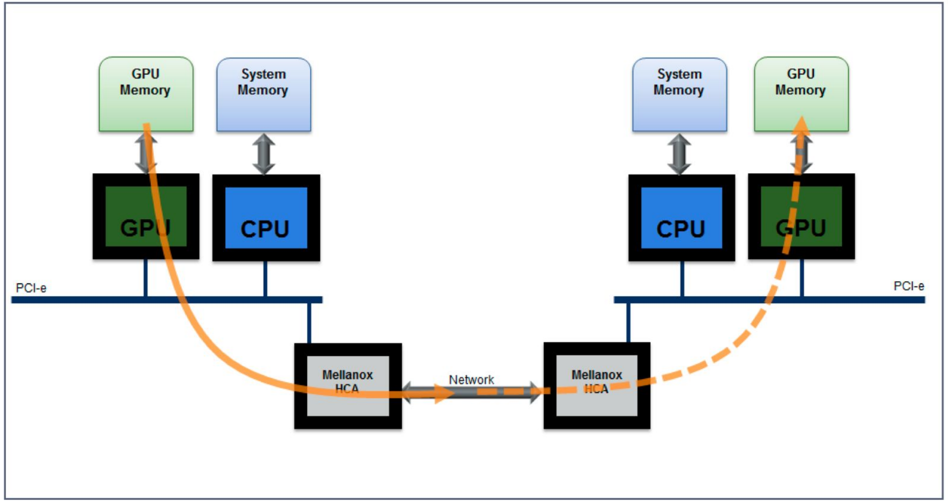


Figure 1. GPUDirect RDMA communication model

"Mellanox OFED GPUDirect RDMA Software Product Brief" (mellanox.com)

- GPUDirect RDMA enables GPUs on different compute nodes to share data without copying the data first to the host CPU
- Removing CPU host memory from the data motion path improves performance due to reduced number of trips along PCIe
- Implemented in network drivers and kernel module - no user intervention required
- Most CUDA-aware MPI implementations already "do the right thing" to take advantage of GPUDirect RDMA

CUDA-aware MPI

- Cray MPI for Perlmutter is "CUDA-aware": the programmer may put pointers to GPU memory in many MPI function calls
- CUDA-aware programming has performance and convenience features:
  - No manual cudaMemcpy() required before calling MPI function
  - MPI library may avoid cudaMemcpy() altogether and use GPUDirect RDMA where appropriate
- Relying on CUDA-aware MPI also has pitfalls
  - Code may crash if running on a system which does not have CUDA-aware MPI

without CUDA-aware MPI

```
cudaMemcpy(buf_h, buf_d, size, cudaMemcpyDeviceToHost);
MPI_Send(buf_h, size, MPI_CHAR, 1, 100, MPI_COMM_WORLD);
```

with CUDA-aware MPI

```
MPI_Send(buf_d, size, MPI_CHAR, 1, 100, MPI_COMM_WORLD);
```

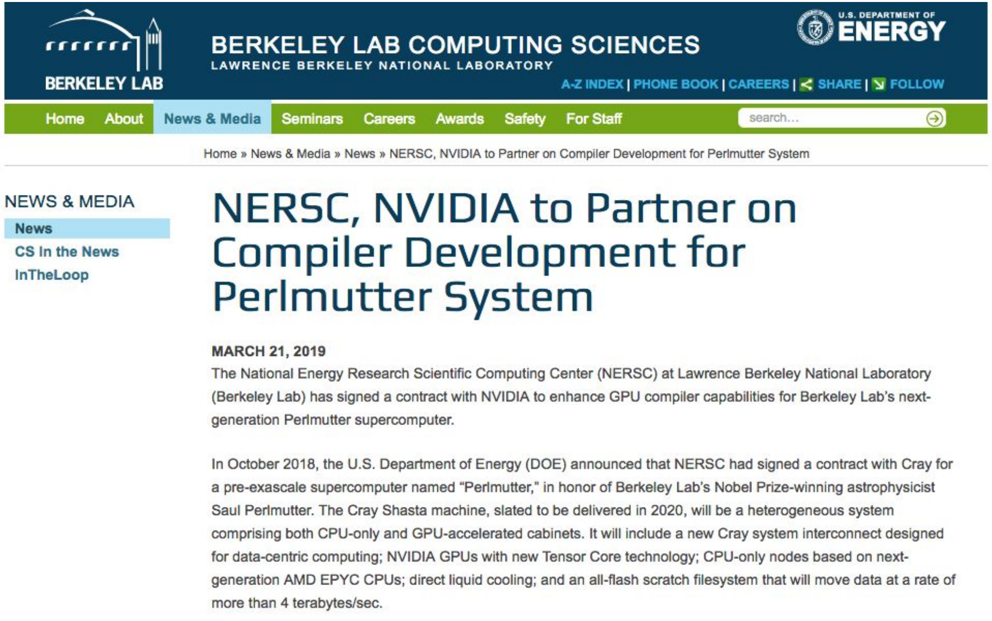
CUDA Unified Memory

- GPU compute nodes on Perlmutter support CUDA Unified Memory
  - CPU and GPU see a common address space, can interact with memory without explicit copies between CPU <-> GPU
  - CUDA runtime automatically migrates unified memory between CPU <-> GPU
- UM provides **convenience** and **consistency**, but not always **performance**

```
cudaMallocManaged(&x, ...);
gpu_kernel<<<nblk, blksz>>>(x, ...);
cpu_function(x, ...);
```

OpenMP NRE partnership with NVIDIA

- Agreed upon subset of OpenMP features to be included in the NVIDIA HPC SDK compiler
- OpenMP test suite created with micro-benchmarks, mini-apps, and the ECP SOLLVE V&V suite
- 5 NESAP application teams partnering with NVIDIA to add OpenMP target offload directives
- NVIDIA HPC SDK compiler versions >= 20.11 include the OpenMP offload capability developed as part of this NRE


BERKELEY LAB COMPUTING SCIENCES
LAWRENCE BERKELEY NATIONAL LABORATORY

Home » News & Media » News » NERSC, NVIDIA to Partner on Compiler Development for Perlmutter System

NERSC, NVIDIA to Partner on Compiler Development for Perlmutter System

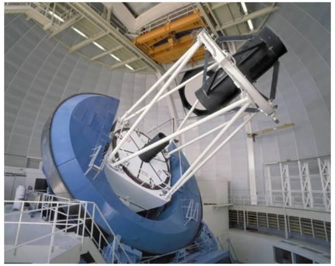
MARCH 21, 2019

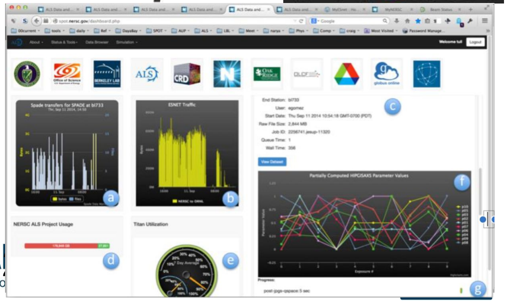
The National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory (Berkeley Lab) has signed a contract with NVIDIA to enhance GPU compiler capabilities for Berkeley Lab's next-generation Perlmutter supercomputer.

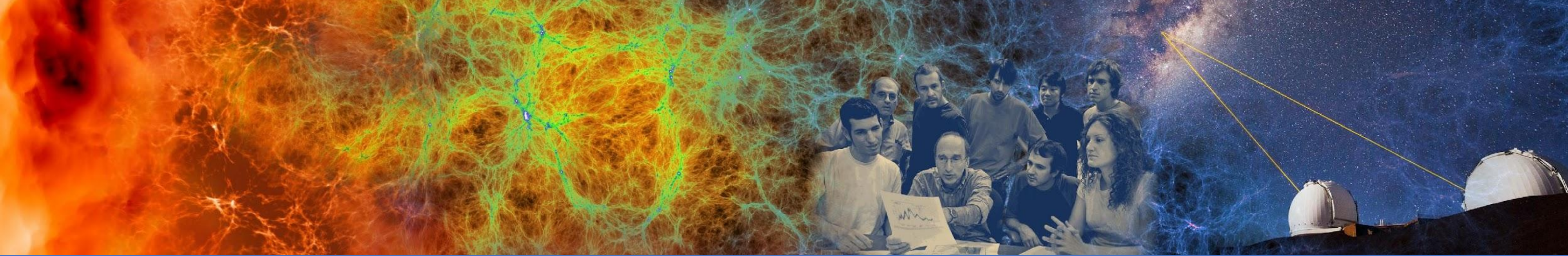
In October 2018, the U.S. Department of Energy (DOE) announced that NERSC had signed a contract with Cray for a pre-exascale supercomputer named "Perlmutter," in honor of Berkeley Lab's Nobel Prize-winning astrophysicist Saul Perlmutter. The Cray Shasta machine, slated to be delivered in 2020, will be a heterogeneous system comprising both CPU-only and GPU-accelerated cabinets. It will include a new Cray system interconnect designed for data-centric computing; NVIDIA GPUs with new Tensor Core technology; CPU-only nodes based on next-generation AMD EPYC CPUs; direct liquid cooling; and an all-flash scratch filesystem that will move data at a rate of more than 4 terabytes/sec.

Analytics and Workflow Integration

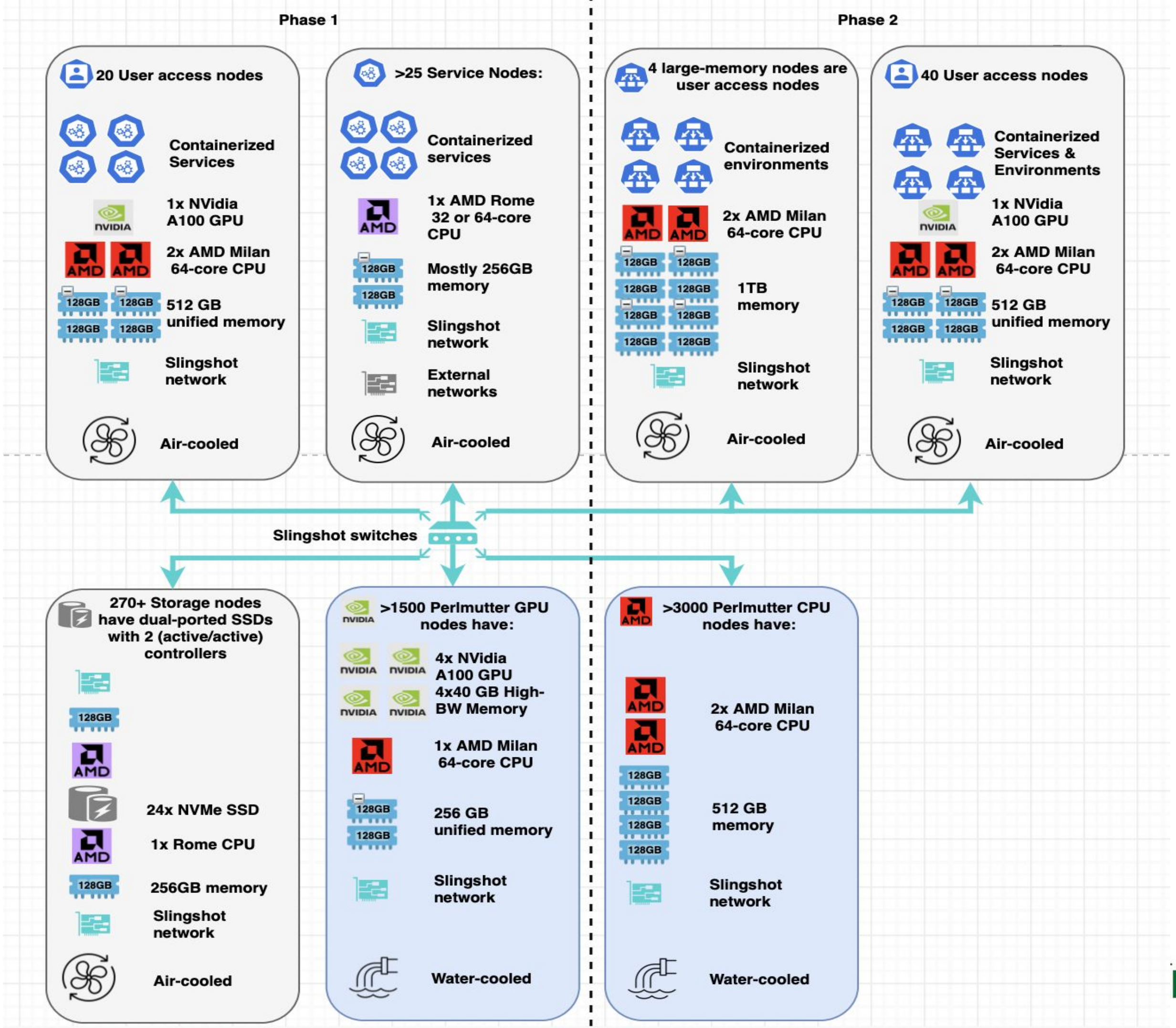
- Software
  - Optimized analytics libraries, includes Cray Analytics stack
  - Collaboration with NVIDIA for Python-based data analytics support
  - Support for containers
- Perlmutter will aid complex end-to-end workflows
  - Slurm co-scheduling of multiple resources and real-time/deadline scheduling
  - Workflow nodes: container-based services
    - Connections to scalable, user workflow pool (via Spin) with network/scheduler access
  - High-availability workflow architecture and system resiliency for real-time use-cases







What is in Perlmutter’s future?



NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

Summary

- Perlmutter is a heterogeneous CPU+GPU system designed to accelerate the diverse data-centric and computational workflows for thousands of NERSC users
- First phase of Perlmutter with all ~6000 NVIDIA A100 GPUs has been delivered in NERSC's data center and is undergoing integration and testing
- The system will support a wide range of programming languages and models, ensuring that its broad workload will be able to use its GPUs effectively